REMARKS/ARGUMENTS

Reconsideration of this application is respectfully requested.

The Examiner's attention is drawn to an anomaly with respect to the partially initialed Form PTO-1449 attached to the Office Action (but not referenced with any check mark on the Office Action Summary page). As filed October 29, 2001, applicant's IDS included three cited US patents and seven publications. A copy of each of these documents was attached (as evidenced by an attached copy of the USPTO postcard receipt acknowledging receipt of all these materials on October 29, 2001).

However, the copy of the Form PTO-1449 returned with the Office Action of 08/30/2004 bears the Examiner's initials opposite only three of these documents (one of the US patents and two of the non-patent publications). Accordingly, a fresh PTO-1449 is attached including again the items that have not yet been initialed – and a fresh copy of all of the non-patent publications is also attached just in case the earlier copies were somehow lost at the USPTO.

In addition, attached hereto is a copy of a UK Search Report and each newly cited reference cited therein. These items are also included on the new Form PTO-1449.

In view of the new citations now being supplied, the IDS fee for this stage of prosecution is also attached. The Examiner's attention to all these references and consideration of them and return of a fully initialed Form PTO-1449 is respectfully requested.

In response to the formality objections, the new claim set submitted herewith has been drafted in keeping with all US claim formatting requirements.

This invention relates to the playing of computer games over a network. Generally, each networked computer terminal in such a game controls at least one entity, which is defined by a data structure containing data used to render the entity in a displayed game environment. This data structure, known as an object, is copied to other terminals in the game, and it is necessary for the originating terminal to update the data in these duplicated objects many times a second in order to ensure that every terminal is displaying the same game environment. While computers are becoming faster and more powerful, networks are often still sluggish and congested. There must therefore be a compromise between keeping the game data current and not sending too many data updates.

As described in the application, it is known to use techniques such as Position History Based Dead Reckoning (PHBDR) to reduce the amount of data that needs to be sent. For each duplicated object stored on it, a terminal uses the last few updates to extrapolate data and uses this data to render and display the game environment. The terminal that stores the originating object extrapolates this same data, and when the extrapolated data and the actual data of the originating object are too different from each other (i.e., if an error value obtained from a comparison of the two data sets is larger than a pre-set error tolerance) an update is sent to the duplicated object, which uses the new data to correct its own data. Techniques such as this reduce network traffic. However, for each player added to a game the increase in the number of updates required increases rapidly. For example, a twenty-player game, each player controlling one entity, has approximately four times as many duplicated objects to be updated as a ten-player game. Since each terminal must update each object several times a second, even games using

PHBDR have an upper limit on how many players can join before the quality of the game is degraded by packet loss and high latency caused by lack of necessary bandwidth.

The present invention improves upon techniques such as PHBDR by introducing a concept of relevance between entities. This may be a Euclidean distance between their positions, it may be a line-of-sight consideration, and so on. A terminal considers how relevant a particular remotely controlled entity is to the entity that it locally controls in order to obtain a relevance value. This value is used to influence the error tolerance used in PHBDR. A high relevance value gives a low error tolerance, and a low relevance value gives a high error tolerance. This means that a terminal sends updates infrequently to terminals controlling entities that are a long way away from or are unseen by the entity controlled by that terminal. Conversely, a terminal sends updates more frequently to terminals controlling entities that are close to and/or visible to the entity controlled by that terminal. This system ensures that a terminal receives frequent updates for duplicated objects representing entities that can affect that terminal's entity in the near future, so that these entities are accurately represented. The less relevant an entity is, the more inaccurate its data can be, and so the terminal receives updates less frequently for duplicated objects representing entities that are less relevant to its own entity.

Using this system, a game can contain many more entities than before. In a large game it is likely that entities will be scattered over a large area, meaning that many entities will not be relevant to each other. Thus new players can join without substantially increasing the amount of data that needs to be sent over the network.

Original claims 1 to 23 have been cancelled in favor of new claims 24-53.

Claim 24 claims a computer network (e.g., the Internet 107 or LAN 211) having a plurality of terminals (e.g., terminals 101, 102 and 103) each having a processor (e.g., CPU 301), a memory (e.g., memory 304), a manual input (e.g., mouse 210) and a network connection (e.g., modem 310 or network card 309), wherein each of said terminals executes instructions to define a shared virtual environment (such as simulation application 803). Each of said instructions includes a local object (such as duplicate master 805) defining a local entity (e.g., aircraft 1001), said object including data (e.g., positional data, as described at paragraph 35) comprising attributes of said entity (e.g., position, as described at paragraph 35), wherein said entity is perceived by a user as being controllable within said shared virtual environment in response to manual control that changes said data. The local object is duplicated on other network terminals as a duplica (e.g., duplica 1004, see paragraph 70).

Each terminal predicts the data of its duplicas (e.g., position history based dead reckoning, see paragraph 40), and each terminal modifies the predicted data of its duplicas in response to receiving updates from the duplicas' originating terminals (see paragraph 64). Each originating terminal sends updates to specific destination terminals in dependence on an assessment of update necessity (e.g., a comparison between actual and predicted data, see paragraph 77), wherein said assessment includes a measurement of relevance between a first entity and a second entity (e.g., a distance measurement, see paragraph 78), said first entity being defined by the local object at said originating terminal and said second entity being defined by a local object at the destination terminal.

Claim 34 claims a method and claim 44 claims a computer-readable medium, both having limitations analogous to those of claim 24.

The rejection of claims under 35 U.S.C. §102 based on alleged anticipation by Addink '477 and the rejection of claims under 35 U.S.C. §103 as allegedly made "obvious" based on Addink '477 in view of Singhal (or Li '316) are all respectfully traversed.

Addink '477 also deals with the problem of multi-player games, where each computer has a game object that is rendered on every other computer. The present invention, however, deals with bandwidth issues, whereas Addink '477 addresses the problem of latency, where a packet of data takes a certain amount of time to travel across a network and therefore is already out of date when it reaches its destination.

Addink '477 discloses a method of time-stamping data packets at the sending computer. A receiving computer uses the timestamp to predict the position of a game object at the current time based on the position contained in the data packet (column 6, line 61 to column 7, line 2). The problem with this approach is that computer clocks are not synchronized. Addink solves the problem by setting a base clock offset at a receiving computer for each of the other computers (column 3, lines 1 to 12). This offset is added onto the timestamp of a data packet before predicting the position of the object (column 6, lines 56 to 65). The receiving computer periodically compares the difference between its own time and the timestamp with the base offset, and adjusts the offset if necessary (column 5, lines 56 to 67, and column 6, lines 27 to 46).

Thus Addink predicts and updates the data of game objects, but this update is not dependent upon an assessment of update necessity and there is no measurement of relevance.

Addink does make a comparison between time values, but the comparison only leads to an adjustment of the base offset. There is no comparison in Addink between the data of two game objects. Thus Addink does not disclose making an assessment of update necessity, nor measuring relevance between entities defined by the objects, as required by the new independent claims, and the new independent claims are thus not taught or suggested by Addink.

In particular, the present invention addresses the problem of lack of bandwidth. In that situation, a method that sends fewer data packets is an advantage. Conversely, Addink deals with latency, and presents a method that allows a receiving computer to estimate the actual time of sending of a packet and adjust the data accordingly. There is no teaching in Addink of how to remedy a lack of bandwidth or how to send fewer data packets, and thus a skilled person would have no reason to look to Addink to solve bandwidth problems.

It is acknowledged that Position History Based Reckoning (PHBDR) is disclosed in Singhal's "Effective Remote Modelling in Large Scale Distributed Simulation and Visualization Environments", as is discussed in the application. However, the amended claims present a system that is an improvement over protocols such as PHBDR, where the data in two objects are compared by the applicant to produce a new relevance value that dynamically alters the error threshold used in PHBDR. There is nothing in Singhal that teaches such a step, since Singhal is concerned with analyzing PHBDR to determine its efficiency rather than with improving it (see the Abstract). Thus the new independent claims are novel and not obvious vis-à-vis the disclosure in Singhal.

Li '316 discloses a system of pre-fetching content from a server based on an optimum bandwidth allocation (see the Abstract). This document does not discuss updating duplicated objects in a shared virtual environment. The new independent claims are thus also patentably distinct vis-à-vis Li '316.

Accordingly, this entire application is now believed to be in allowable condition and a formal Notice to that effect is respectfully solicited.

Respectfully submitted,

NIXON & VANDERHYE P.C.

By: Larry S. Nixon
Larry S. Nixon
Reg. No. 25,640

LSN:vc
1100 North Glebe Road, 8th Floor
Arlington, VA 22201-4714
Telephone: (703) 816-4000
Facsimile: (703) 816-4100